

Zusammenfassung DASB

Data Science Basics

Maurin D. Thalmann

24. Januar 2020

Inhaltsverzeichnis

1 Learning Objectives	2
2 Introduction to Data Science	3
2.1 Introductory Stizzle	3
2.2 A Global View on Data Science	3
2.3 Different Concepts in Data Analytics	4
2.3.1 1993 - Online Analytical Processing (OLAP)	4
2.3.2 1998 - Fuzzy Data Analysis	4
2.3.3 2005 - Data Mining	5
2.3.4 2013 - Predictive Modeling	5
2.3.5 2013 - Data Driven Decision Making	5
3 Intro to R and Exploratory Data Analysis (EDA)	6
3.1 Intro to R	6
3.1.1 Basic Operation in R	6
3.2 What is EDA	6
3.2.1 EDA process „in practice“	7
3.3 Data Import / Transform / Link	7
3.3.1 Data Import	7
3.3.2 Data Transform (example dataset)	7
3.3.3 Data Transform	8
3.3.4 Data „Linkage“	8
3.3.5 Applying data loading/transforming/linking	9
4 ggplot2 and its usage in EDA	9
4.1 ggplot2 - Intro	9
4.2 ggplot2 - layer after layer	10
4.3 Elements of grammar of graphics	10
4.4 Aesthetics	10
4.5 Geoms	10
4.6 Stats & Scales	11
4.7 ggplot2 - Initial Example	11
4.8 Coordinate System & Faceting	12
4.9 ggplot2 - Second Example	12
5 Shiny	13
6 Intro to Forecasting	13
6.1 Indication	13
6.2 Linear Modeling	13
6.2.1 Modeling	13
6.2.2 Model Identification	14
6.2.3 Prediction	14
6.2.4 Model Evaluation	14
7 Forecasting Models: Usage / Application	15
7.1 Forecasting vs. Interpretation	15
7.2 Performance vs. Interpretability	15
7.3 Parametric vs. non-Parametric models	15
7.3.1 Pros and Cons (Param vs. non-Param)	16
7.4 General approach to ML / Data Science	16
7.5 An example (yup, that's the title)	16
7.5.1 Step 1: Load the Data	17
7.5.2 Step 2: Preprocess the data	17
7.5.3 Step 3: Derive features (predictors)	17
7.5.4 Step 4: Build and Train the Model	18

7.5.5 Step 5: Improve the model	20
7.6 How to choose the right model (ML view)	20
8 Ethics in Data Science	21
8.1 Data Ethics	21
8.2 Big Data Analytics and Data Science	21
8.3 Possible problems with data science	21
8.4 Ethical concerns about personal data	22
8.5 Conditions for processing personal data	22
8.6 Purpose Limitation	22
8.7 Data Minimization: Collection & Retention	22
8.8 Accuracy	23
8.9 Rights of Individuals	23
8.10 Security	23
8.11 Accountability and Governance	23
8.12 Data Controllers & Data Processors	23
8.13 Key Recommendations	24
8.13.1 Anonymization	24
8.13.2 Privacy Notices	24
8.13.3 Privacy Impact Assessment	24
8.13.4 Privacy by Design	24
8.13.5 Ethical Approaches	24
8.13.6 Algorithmic Transparency	24

1 Learning Objectives

Introduction to Data Science

- Define the term „Data Science“ and differentiate it from similar concepts such as big data, data mining, OLAP, predictive modeling, data analysis
- Describe why and how value is created from data, especially in form of processes
- Describe the skills and activities of a data science professional
- Describe possible applications of data science

Intro to R and Exploratory Data Analysis (EDA)

- Define the term „Exploratory Data Analysis“ and explain why it is a fundamental part of making data science
- Give a short introduction to R and present its main features
- Indicate the key actors and actions that compose EDA and how they are related to the data sources considered
- Indicate different ways of importing data into R for analysis and how to check for their quality
- Describe the main operations available in R (through libraries) to manipulate data and prepare it for the analysis

ggplot2 and its usage in EDA

- Define the role of graphical representations in the „Exploratory Data Analysis“ and explain why it is a fundamental tool to make it effective
- Give a short introduction to ggplot2 and its features
- Dddescribe the main way of working of ggplot2 and how to compose complex graphs (e.g.: representing more than 3 variables in a single graph, for multidimensional data)
- Explain the role of the facets and how they can support the representation of „partial“ relationships
- Indicate the role of the coordinates in the graph usage for EDA

Intro to Forecasting

- Understand and explain the concept of forecasting
- Present the basic approach, either using the theory or by example
- Present a linear model and describe its basic characteristics
- Indicate how EDA can support and scaffold the model identification
- Apply basic forecasting in R, given the datasource

Forecasting Models: Usage / Application

- Presents the limits of the linear models
- Define the way of extending linear models
- Justify forecasting & interpretation (phenomena comprehension) in data science, also by presenting examples
- Present advantages and limits of the defined extensions
- Presenting Parametric and Nonparametric idea in model definition (ML)
- Indicate how EDA can support and scaffold the model identification
- Apply more advanced forecasting in R, given the datasource

Ethics in Data Science

- List ethical issues with big data analytics
- Know the conditions for processing of personal data
- Know the Rights of Individuals regarding personal data processing
- Provide recommendations for more ethical data science
- Being able to analyze a given case and provide conclusion on the ethical aspects involved

2 Introduction to Data Science

2.1 Introductory Stizzle

- More is different - with different quantity comes different quality, although „more“ is always dependent on the point of view.
- Data volume nowadays is exploding (276.12 billion GB of digital data)
- Data Science tries to close the gap between „Data available to an organization“ and „Percent of data an organization can process“.

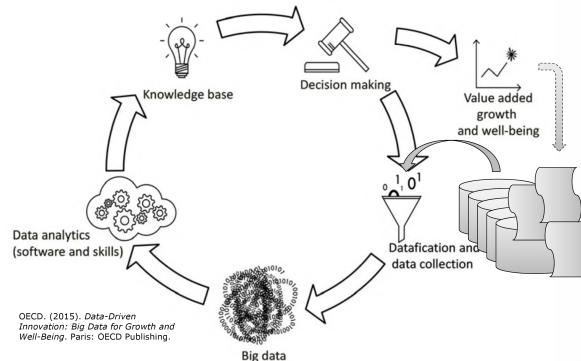


Abbildung 1: Beautiful illustration of the classic data value cycle

„**Data Science** is the extraction of actionable knowledge directly from data through a discovery, or hypothesis formulation and hypothesis testing“. Data Scientists generate knowledge from big data.

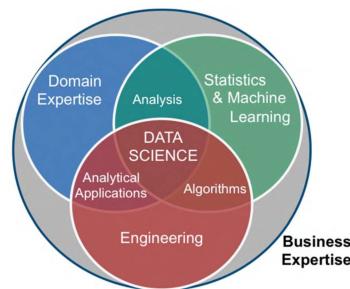


Abbildung 2: Skills needed in Data Science

2.2 A Global View on Data Science

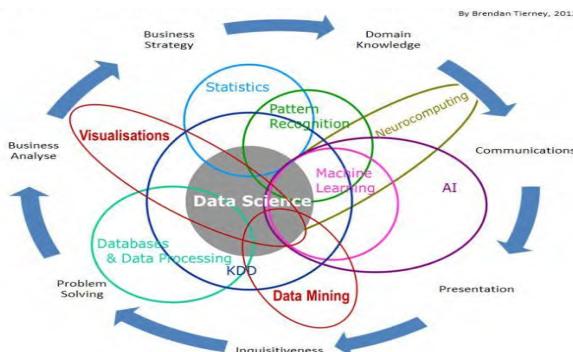


Abbildung 3: Data Science is multidisciplinary

- **AI**: programs that perform tasks resembling humans (learn and reason)
- **ML**: algorithms to learn from that data without explicit programming
- **DL**: subset of ML using artificial neural networks for treating vast amount of data (big data)
- **Data Science**: spans the collection, management, analysis and interpretation of large amounts of data with a wide range of applications → make informed decisions based on what was learned
- **EDA** (exploratory data analysis): extract insight from data (outside AI, human based)

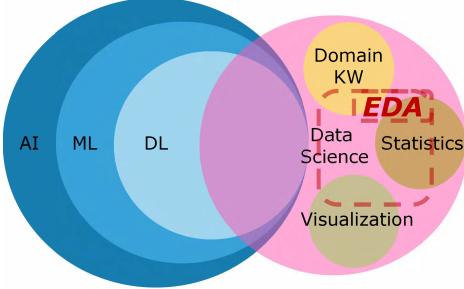
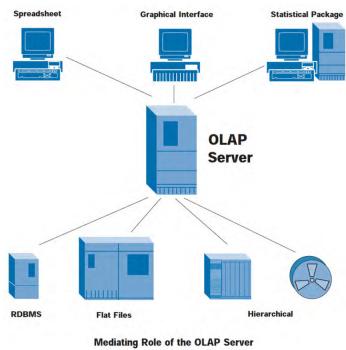


Abbildung 4: AI / ML / DL and Data Science

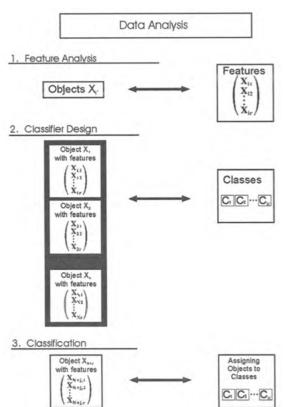
2.3 Different Concepts in Data Analytics

2.3.1 1993 - Online Analytical Processing (OLAP)



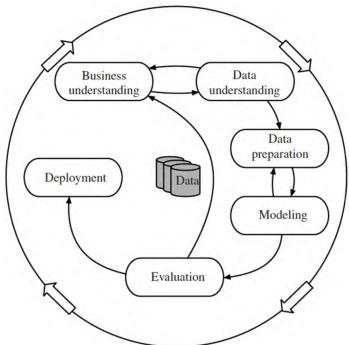
- It's online, not batch (interactive, not programmed in COBOL)
- The OLAP system should access the data required to perform the indicated analysis
- OLAP tools empower useranalysts to easily perform multi-dimensional analysis, which previously have been avoided because of their perceived complexity.

2.3.2 1998 - Fuzzy Data Analysis



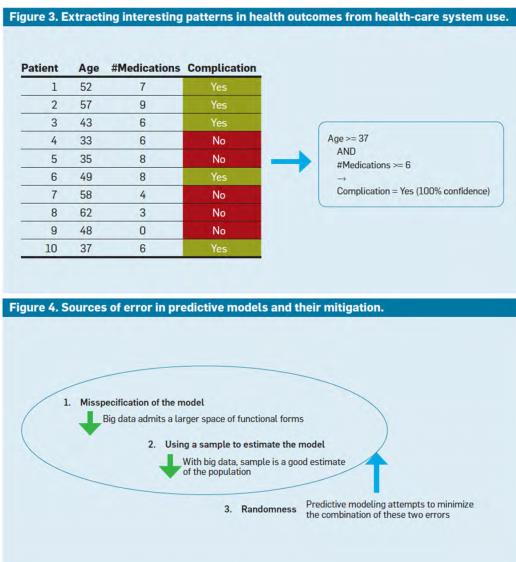
- Data analysis can be defined as **search for structure** in data
- In data analysis, objects are considered which are described by some attributes
- Most of the traditional methods for data analysis assume that patterns to be detected are two-valued
- Whenever this is not the case, the relationship between data and classes becomes gradual
→ Fuzzy Classification

2.3.3 2005 - Data Mining



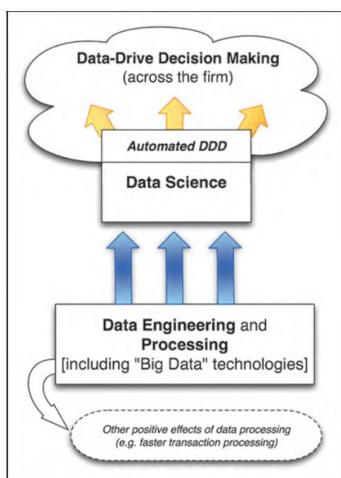
- Very similar concept to data science
- Machine Learning (modeling) is the technical core of practical data mining applications
- Data Mining is a business process related to *value* (finding the metaphorical gold nugget)
- The lifecycle of a data mining project is defined by the CRISP-DM reference model

2.3.4 2013 - Predictive Modeling



- A common epistemic requirement in assessing whether new knowledge is actionable for decision making in its *predictive power*, not just its ability to explain the past.
- The requirement on predictive accuracy on observations that will occur in the future is a key consideration in data science.

2.3.5 2013 - Data Driven Decision Making



- Data science involves much more than just data-mining algorithms.
- Successful data scientists must be able to view business problems from a *data perspective*.

3 Intro to R and Exploratory Data Analysis (EDA)

3.1 Intro to R

- R ist a software and programming language for statistical applications and graphics
- It's an implementation of the programming language S, with some extensions
- It provides functions for linear/non-linear modelling, classical statistical tests, time-series analysis, classification, clustering etc.
- RStudio is an integrated development environment (IDE) for R

3.1.1 Basic Operation in R

- Assignment:
`my_var <- 46, nr <- 38`
(You can also use `copy my_var = 46`, but it is not suggested)
- Printing (explicit and implicit): every line with only a name of an object
Implicit: `my_var`
Explicit using functions like `print(print(my_var))`
- Arithmetic operators:
`4 + 5`
`2 ^ 3 - 7 %% 3 - 2 ^ (3 - 7 %% 3)` etc. etc. etc.
- Control flow, conditionals, loops can be used as in any other language
`for (i in 1:10) print(i)`
`while (nr < 42) print(,We have "+str(nr)+,, : still space..."); nr=nr+1`

3.2 What is EDA

- Exploratory Data Analysis is the
 - critical process of performing initial investigations on data so as to
 - * discover patterns
 - * spot anomalies
 - * test hypothesis
 - * check assumptions
 - with the help of **summary statistics** and **graphical representations**

The use of a statistical model is not mandatory, in this step, even if it is normally adopted to test hypotheses and check assumptions. For anomalies and patterns discovery, correlations (represented into the correlation matrix) and distribution (represented with box and whisker diagrams) are the usual approaches.

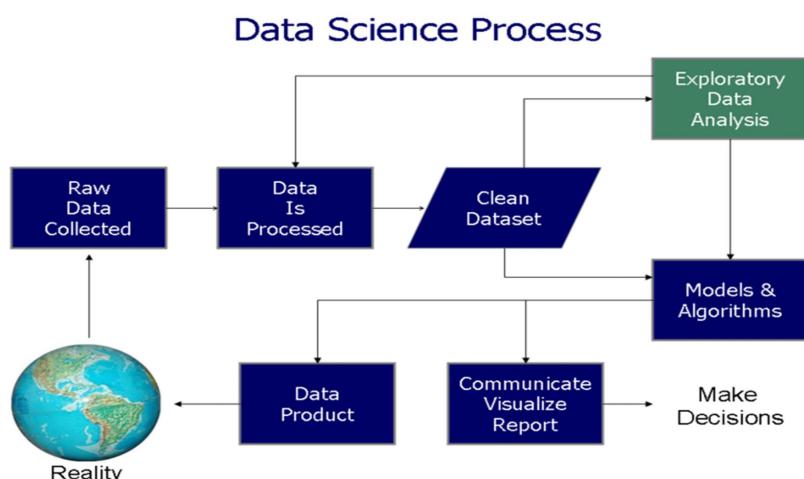


Abbildung 5: EDA in the context of Data Science

3.2.1 EDA process „in practice“

- EDA is an iterative cycle
- Process:
 - Generate questions about the data
 - Search for answers by visualizing, transforming and modelling the data
 - Use the insights to refine the questions and/or generate new questions
- Start with many ideas...
- some will be evidently useless and other will end up in a dead end
- a few of them will help to have a better understanding of the data/process
- repeat with generating other ideas until the insight is enough
- then you can apply the model to the data for forecasting and process explanation
- EDA also allows you to verify/test the quality of your data

3.3 Data Import / Transform / Link

3.3.1 Data Import

- R provides importing functions, depending on the format of the data source:
 - CSV / TSV / delimited texts: `read_csv()` & `read_csv2()` (semicolon), `read_tsv()`, `read_delim()`
 - Fixed width columns: `read_fwf()` & `read_table()`
 - Logs (e.g. webserver logs): `read_log()`

These are all similar in terms of parameters, examples:

```
1 read_delim(file , delim , quote = "\\"", escape_backslash = FALSE, escape_double = TRUE,
2 col_names = TRUE, col_types = NULL, locale = default_locale() , na = c("", "NA"),
3 quoted_na = TRUE, comment = "", trim_ws = FALSE, skip = 0, n_max = Inf, guess_max =
```

3.3.2 Data Transform (example dataset)

Example dataset loaded from: *Auto MPG Data Set*

```
1 > library(tidyverse)
2 > DS <- read.table("C:/Users/JumpStart/Downloads/auto-mpg.data",
3 quote="\\"", comment.char="")
4 > names(DS) <-
5 c("mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration",
6 "model year", "origin", "car name")
7 > DS <- as_tibble(DS); attach(DS); View(DS)
```

3.3.3 Data Transform

- Filter rows with `filter()`
 - subset observations based on their values
`filter(DS, weight > 3500) or filter(DS, DS$weight > 3500)`
- Arrange rows with `arrange()`
 - changes the order of rows
`arrange(DS, desc(DS$acceleration))`
- Select columns with `select()`
 - Subset columns with position and/or negative condition (by name or index)
`select(DS, -1)`
`select(DS, 2, 3, 4)`
`select(DS, c(mpg, 5))`
- Add new variables with `mutate()`
 - Allow to create new dimension(s)
`mutate(DS, a4w = acceleration/weight, pre75 = as.numeric(DS$model year' < 75))`
- Grouped summaries with `summarise()`
 - Allows to compute aggregate metrics
`summarise(group_by(DS, 'model year'), mean_display = mean(displacement), sd_displ = sd(displacement), n=n())`
- All verbs work similarly:
 - The first argument is a dataframe
 - The subsequent arguments describe what to do with the data frame, using the variable names (without quotes)
 - The result is a new data frame

3.3.4 Data „Linkage“

Some prices of cars of the 70's: *70's Cars*.

This data is imported into a new tibble and we will try to use it for creating another View for exploring the relationship between cost, year, power and mpg.

```

1 > car_cost <- read_delim("C:/Users/JumpStart/switchdrive/__/DASB/SW02/car_197x_costs_2.csv",
2   delim = "_",
3   escape_double = FALSE,
4   na = "NA",
5   trim_ws = TRUE,
6   col_types = cols('Car Name' = col_factor(),
7     Price = col_character(),
8     Location = col_character(),
9     'Matriculation Year' = col_integer()))
10 > car_cost <- as_tibble(car_cost)
11 > car_cost <- mutate(car_cost, Location = sapply(Location, as.factor))
12 > car_cost <- mutate(car_cost, Price = as.integer(str_replace(str_sub(Price, 2, -1), ",","")))
13 > View(car_cost)

```

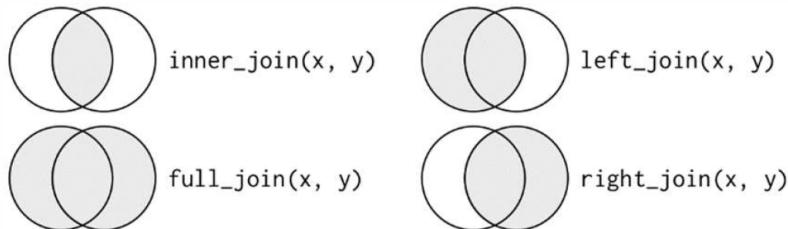


Abbildung 6: R functions to link data frames with each other

There are different types of R functions for linking tibbles/dataframes (~datasets)

- **Mutating** joins allows to combine variable from two tables
 - `inner_join(x, y, by = keys)`
matches pairs of observations whenever their keys are equal
 - Outer join keeps observations that appear in at least one of the tables. There are three types of outer joins:
 - * `left_join(x, y, by = keys)`
keeps all observations in x
 - * `right_join(x, y, by = keys)`
keeps all observations in y
 - * `full_join(x, y, by = keys)`
keeps all observations in x and y
- **Filtering** joins allow to restrict the set of row from the 1st table
 - `semi_join(x, y, by = keys)`
keeps all observations in x that have a match in y
 - `anti_join(x, y, by = keys)`
drops all observations in x that have a match in y
- **Set Operations** (work on complete rows, comparing each variable values. Inputs needs the same variables, for treating observations like sets)
 - `intersect(x, y)`
return only observations in both x and y
 - `union(x, y)`
return unique observations in x and y
 - `setdiff(x, y)`
return observations in x, but not in y

3.3.5 Applying data loading/transforming/linking

- Find one or more data sources you are interested into (UCI ML Repository / Kaggle)
- Start the EDA „mindset“:
 1. What I believe about the data / what I would like to discover
→ Make hypotheses and formulate assumptions
 2. Think how this can be supported by data → experiment
 3. Run the experiment to verify
 4. Acquire new knowledge that can help you improve step 1
- Repeat the process at least 3 times
(Optional: short report of the process you followed, document it in a short presentation)

4 ggplot2 and its usage in EDA

4.1 ggplot2 - Intro

- Used with the library „ggplot2“
`library(ggplot2)`
- Simple usage example:
`plot <- ggplot(data = mpg, mapping = aes(x = mpg$hwy))
plot + geom_histogram()`
- ggplot 2 package:
 - produces layered statistical graphics.
 - uses an underlying „grammar“ to build graphs layer-by-layer rather than providing premade graphs
 - is easy enough to use without any exposure to the underlying grammar, but is even easier to use once you know the grammar
 - allows the user to build a graph from concepts rather than recall of commands and options

4.2 ggplot2 - layer after layer

- The initial call to ggplot does create the graph, but no content:

```
ggplot(data = mpg)
ggplot(data = mpg, mapping = aes(x = mpg$hwy))
ggplot(data = mpg, mapping = aes(x = mpg$hwy, y = mpg$displ))
```
- We need to add a layer by using + (or collect the partial graph in an object)

```
p <- ggplot(data = mpg)
p + geom_histogram(mapping = aes(x = mpg$hwy))
```
- Parameters set into the ggplot() call are inherited by every other layer, but can be overridden, if needed

```
p <- ggplot(data = mpg)
p <- p + geom_histogram(mapping = aes(x = mpg$hwy))
p
p + geom_violin(mapping = aes(y = mpg$displ))
```

4.3 Elements of grammar of graphics

- **Data:** variables mapped to aesthetic features of the graph.
- **Geoms:** objects / shapes on the graph.
- **Stats:** statistical transformations that summarize data (e.g. mean, confidence intervals).
- **Scales:** mappings of aesthetic values to data values. Legends and axes visualize scales.
- **Coordinate systems:** the plane on which data is mapped on the graphic.
- **Faceting:** splitting the data into subsets to create multiple variations of the same graph (paneling).

4.4 Aesthetics

- Visual properties of objects on the graph (depends on the geom*)
- Commonly used aesthetics:
x positioning along x-axis
y positioning along y-axis
color color of objects; for 2D-objects: the color of the object's outline (compare to fill below)
fill fill color of objects
linetype how lines should be drawn (solid, dashed, dotted etc.)
shape shape of markers in scatter plots
size how large objects appear
alpha transparency of objects (value from 0 [transparent] to 1 [opaque])

4.5 Geoms

- Geometric shapes produced for the plot.
 - `geom_bar()`: bars with bases on the x-axis
 - `geom_boxplot()`: boxes and whiskers
 - `geom_density()`: density plots
 - `geom_histogram()`: histogram
 - `geom_line()`: lines
 - `geom_point()`: points (scatterplot)
 - `geom_rug()`: rug plot (2D display with the two 1D marginal distributions)
 - `geom_smooth()`: smoothed conditional means
 - `geom_text()`: text

4.6 Stats & Scales

- **Stats** → statistically transform data, usually as some form of summary (e.g.: mean, standard deviation, confidence interval etc.). Each stat function is associated with a default geom, so no geom is required for shapes to be rendered.
- **Scales** → control how a plot maps data values to the visual values of an aesthetic
 - `scale_color_manual()`:
define an arbitrary color scale, by specifying each color manually
 - `scale_color_hue()`:
define an evenly-spaced color scale, by specifying a range of hues and the number of colors on a scale
 - `scale_shape_manual()`:
define an arbitrary shape scale, by specifying each shape manually
- Scales can also be applied to axes:
 - `scale_x_continuous()`:
map continuous to visual values - `scale_x_discrete()`
 - `scale_x_date(labels = date_format(,,%m/%d“), breaks = date_breaks(,,2 weeks“))`:
treat x values as dates - `scale_x_datetime()`
 - `scale_x_log10()`:
plot x on log10 scale
 - `scale_x_reverse()`:
reverse direction of x-axis
 - `scale_x_sqrt()`:
plot x on square root scale

4.7 ggplot2 - Initial Example

```
1 ggplot(diamonds, aes(x=carat, y=price, color=cut)) +
2   geom_point() +
3   scale_color_manual(
4     values=c("red", "yellow", "green", "blue", "violet")
5   ) +
6   scale_y_continuous(
7     breaks=c(0,2500,5000,7500,10000,12500,15000,17500),
8     labels=c(0,2.5,5,7.5,10,12.5,15,17.5),
9     name="price (thousands of dollars)"
10 )
```

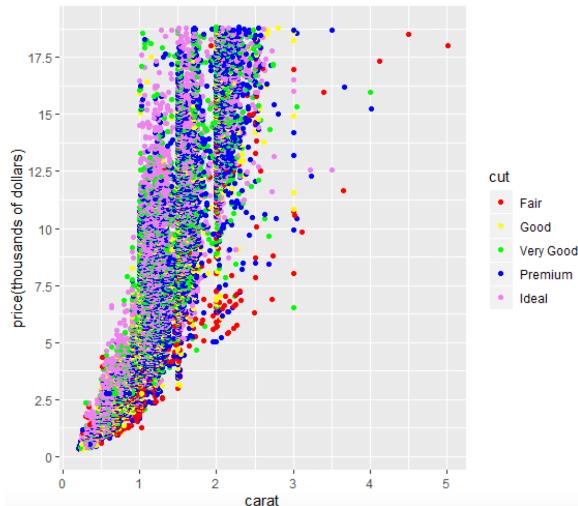


Abbildung 7: Rendered graph for Initial Example

4.8 Coordinate System & Faceting

- **Coordinate system:** set the planes for objects positioning on the plot
- Cartesian is the most diffused and used by default. Other coordinate systems would be:
 - polar
 - flipped Cartesian
 - fixed-ratio Cartesian
 - Map projections (mercator, lagrange)
- **Faceting:** divide the graph into subgraphs (panels), using variable values
 - `facet_wrap(~{vars})`:
create a multirow panel of plots, `{vars}` represents a list of splitting variables, separated by +
 - `facet_grid({vars}~{vars})`:
row-splitting variable before, the column-splitting variable after
- Other aspects include the teming, position adjustment, labels and legends, zooming and the graph savings (`ggsave`)

4.9 ggplot2 - Second Example

```
1 ggplot(diamonds, aes(x=carat, y=price, color=cut)) +
2   geom_point() +
3   facet_grid(clarity~color) +
4   scale_color_manual(values=c("red", "yellow", "green", "blue", "violet")) +
5   scale_y_continuous(breaks=c(0,2500,5000,7500,10000,12500,15000,17500),
6   labels=c(0,2.5,5,7.5,10,12.5,15,17.5),name="price(thousands of dollars)")
```

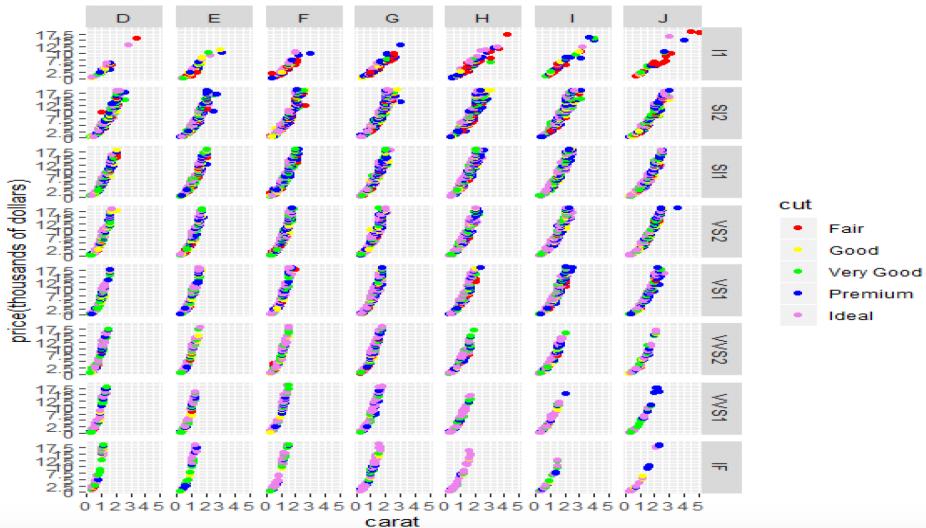


Abbildung 8: Rendered graph for Second Example

5 Shiny

Machi de wenni no Loscht han. (Wenn de Text am Schloss no do esch, hani kei Loscht meh gha)

6 Intro to Forecasting

6.1 Indication

- Forecast: the ability to use a set of variables for predicting another one
- Predictors $\rightarrow F(x) \rightarrow$ Output
- Can assume two forms:
 - Continuous Output \rightarrow Regression
 - Discrete (Categorical) Output \rightarrow Classification
- It uses similar approaches, but with a different result
- The function $F(x)$ is called the model. Can be used for two goals:
 - Pure prediction: care only for the forecast
 - Interpretation: care of the existing relationship, to better understand the phenomena

6.2 Linear Modeling

6.2.1 Modeling

- Simplest form of a model is the Linear Model
 - With one predictor (bidimensional space) represent the equation of a line:
- $$y = F(x) = \beta_0 + \beta_1 \cdot x$$

$$Y \approx \beta_0 + \beta_1 \cdot X$$

$$e_i = y_i - \hat{y}_i$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

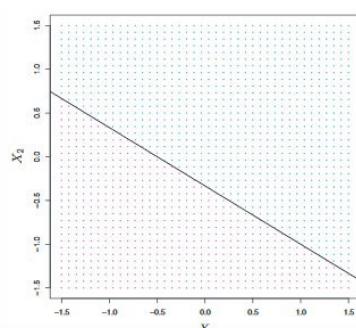
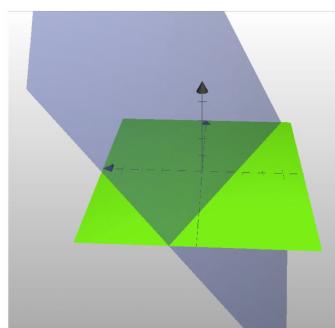
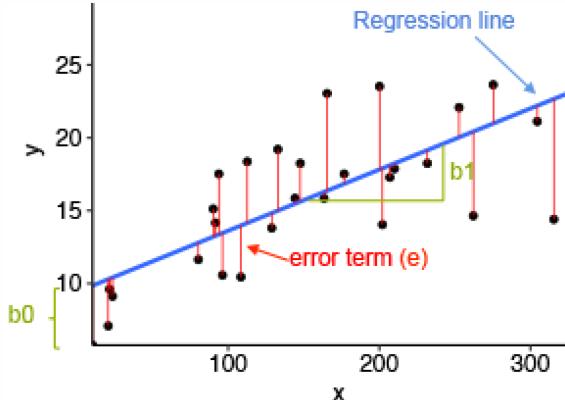
$$\bar{y} \equiv \frac{1}{n} \sum_{i=1}^n y_i$$

$$\bar{x} \equiv \frac{1}{n} \sum_{i=1}^n x_i$$

$$RSS = e_1^2 + e_2^2 + \dots + e_n^2$$

- With more predictors (multidimensional space or Hyperspace), the model can be seen as a hyperplane:

$$y = F(x) = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \dots + \beta_n \cdot x_n$$



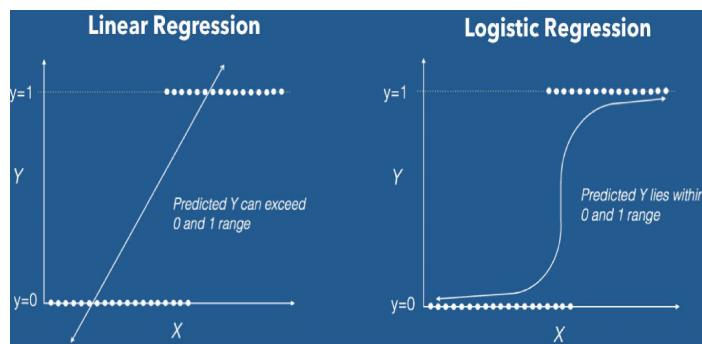
6.2.2 Model Identification

Following steps need to be done

- Define the predictors set and the output variables (also using correlation)
 - $x = \dots, y = \dots$
- Create the model
 - `model <- lm(y ~ x, data = d)`
- Inspect the model
 - Parameters $\beta_0, \beta_1, \beta_2, \dots, \beta_n$
- Verify that the model is „trustable“
 - P-values of each β_a , everyone should be under 0.05 (5%)

6.2.3 Prediction

- It is natural to interpret the model for regression:
 - Given a single set of predictor values X (point) → compute the value of the output (y)
- but what about for classification?
 - We need to define a way to transform the output of the linear model to our output categories
 - This is done by the logistic function
- How to do predictions:
 - `Pred <- predict(model, data = d2)`



6.2.4 Model Evaluation

- We need to evaluate models based on performances
 - Regression → the sum of squared errors (RSS → residual sum of squares)
 - Classification → the misclassification error rate
- Every model with more predictors has a higher predictive power...
 - ...but, we don't want to learn the noise on the data!
- Create an independent test set
 - Use most part (e.g. 70%) of the data for training purposes
 - Use the rest for the model performance estimation → Test Error Rate

7 Forecasting Models: Usage / Application

7.1 Forecasting vs. Interpretation

- Linear model
 - Rigid (strong assumptions about the relationship)
 - * Predictors can only be represented by variables
 - * Predictors do not have interactions (can be analyzed separately)
 - $y = F(X) = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$
- Two extensions:
 - Transformation applied to variables, to be used as predictor and/or output
 - * E.g.: $\log_2(y) = F(X) = \beta_0 + \beta_1 \sin x_1 + \dots + \beta_n e^{x_n}$
 - * Formula $\rightarrow y \sim I(\sin(x_1)) + \dots + I(\exp(x_n))$
 - Interaction between variables
 - * E.g.: $y = F(X) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{1-2} x_1 x_2$
 - * Formula $\rightarrow y \sim x_1 \cdot x_2$

7.2 Performance vs. Interpretability

- Every extension adds more predictive performances, but reduces interpretability.
- Limits (boxes between Input $\rightarrow \dots \rightarrow$ Output):
 - Pure linear model \rightarrow white box
 - Complex extensions \rightarrow black box
 - Intermediate \rightarrow gray boxes

7.3 Parametric vs. non-Parametric models

- All linear models (and extensions) are called **parametric** models:
 - They make assumptions on the data distribution
 - Are bound to the amount they can learn, depending on the fixed number of parameters of the model
 - Other examples of parametric models are:
Logistic Regression, Polynomial Regression and Perception
 - A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a parametric model. No matter how much data you throw at a parametric model, it won't change its mind about how many parameters it needs.
- More complex models from Machine Learning are **non-parametric**:
 - Nonparametric methods are good when you have a lot of data and no prior knowledge, and when you don't want to worry too much about choosing just the right features.
 - The amount of information that the model is able to capture can grow with the increase in the amount of data analysed.
 - Examples of non-parametric models are: k-Nearest Neighbors (kNN), Decision Trees, Support Vector Machines, Neural Networks
 - They are normally more powerful (predictive power), but their interpretation is limited or basically impossible.

7.3.1 Pros and Cons (Param vs. non-Param)

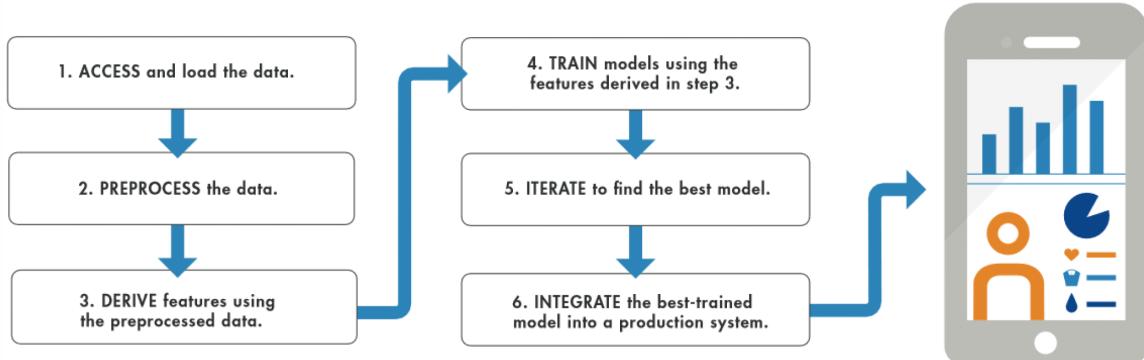
	Benefits	Limits
Param	Simpler: easier to understand and to interpret its results. Fast: faster to learn from data. Less Data: limited training data needed and can work well also with imperfect data fit.	Constrained: By choosing a functional form highly constrained to it. Limited Complexity: more suited to simpler problems. Underfitting: unlikely to match the underlying mapping function, if complex. → BIAS
Non-Param	Flexibility: Capable of fitting a large number of functional forms. Power: No assumptions (or weak assumptions) about the underlying function. Performance: Can result in higher performance models for prediction.	More data: significantly more training data required to estimate the mapping function. Slower: train require more data and then more time, due to the larger number of parameters to estimate. Overfitting: risk of overfitting the training data if not properly constrained. → VARIANCE Explicability: harder to explain why specific predictions are made, if even possible.

Rigidness (Bias) vs. Instability/Sensibility to data (Variance) tradeoff. It depends on:

- the amount of data
- the complexity and the underlying function/relationship to estimate

7.4 General approach to ML / Data Science

Mathwork guides about ML: Guide 1 and Guide 2



7.5 An example (yup, that's the title)

- This example is based on a cell phone health-monitoring app.
- The input consists of three-axial sensor data from the phone's accelerometer and gyroscope.
- The responses (or output) are the activities performed: walking, standing, running, climbing stairs or lying down.
 - **Supervised** = we have the classes
 - **Unsupervised** = we would like to discover some structure, but don't have a real output variable
- Supervised → Classification
 - The trained model (or classifier) will be integrated into an app to help users track their activity levels throughout the day.

The five steps:

- Step 1: Load the data
- Step 2: Preprocess the data
- Step 3: Derive features (predictors)
- Step 4: Build and train the model
- Step 5: Improve the model

7.5.1 Step 1: Load the Data

- To load data from the accelerometer and gyroscope, we need to do the following:
 1. Sit down holding the phone, log data from the phone, and store it in a text file labeled „Sitting“.
 2. Stand up holding the phone, log data from the phone, and store it in a second text file labeled „Standing“.
 3. Repeat the steps until we have data for each activity we want to classify.
- We store the labeled data sets in a text file.
- A flat file format such as text or CSV is easy to work with and makes it straightforward to import data.
- Machine learning algorithms are not smart enough to tell the difference between noise and valuable information.
- Before using the data for training, we need to make sure it's clean and complete.

7.5.2 Step 2: Preprocess the data

- We import the data into our environment and plot each labeled set. To preprocess the data we do the following:
 1. Look for outliers → data points that lie outside the rest of the data
 - We must decide whether the outliers can be ignored or whether they indicate a phenomenon that the model should account for.
 - In our example, they can safely be ignored (it turns out that we moved unintentionally while recording the data).
 2. Check for missing values (perhaps we lost data because the connection dropped during recording).
 - We could simply ignore the missing values, but this will reduce the size of the data set. Alternatively, we could substitute approximations for the missing values by interpolating or using comparable data from another sample (data augmentation).
 3. Remove gravitational effects from the accelerometer data so that our algorithm will focus on the movement of the subject, not the movement of the phone. A simple high-pass filter such as a biquad filter is commonly used for this.
 4. Create a training and test set (Also eventually a train-test-validation, e.g. 70% / 20% / 10%)

7.5.3 Step 3: Derive features (predictors)

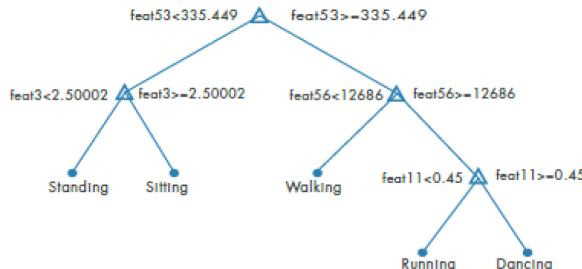
- Deriving features turns raw data into information that a machine learning algorithm can use to learn the model.
- For the activity tracker, we want to extract features that capture the frequency content of the accelerometer data. These features will help the algorithm distinguish between walking (low frequency) and running (high frequency). We create a new table that includes the selected features.
- Derive as many features as you can think of.

Data Type	Feature Selection Task	Techniques
Sensor data	Extract signal properties from raw sensor data to create higher-level information	Peak analysis – perform an fft and identify dominant frequencies Pulse and transition metrics – derive signal characteristics such as rise time, fall time, and settling time Spectral measurements – plot signal power, bandwidth, mean frequency, and median frequency
Image and video data	Extract features such as edge locations, resolution, and color	Bag of visual words – create a histogram of local image features, such as edges, corners, and blobs Histogram of oriented gradients (HOG) – create a histogram of local gradient directions Minimum eigenvalue algorithm – detect corner locations in images Edge detection – identify points where the degree of brightness changes sharply
Transactional data	Calculate derived values that enhance the information in the data	Timestamp decomposition – break timestamps down into components such as day and month Aggregate value calculation – create higher-level features such as the total number of times a particular event occurred

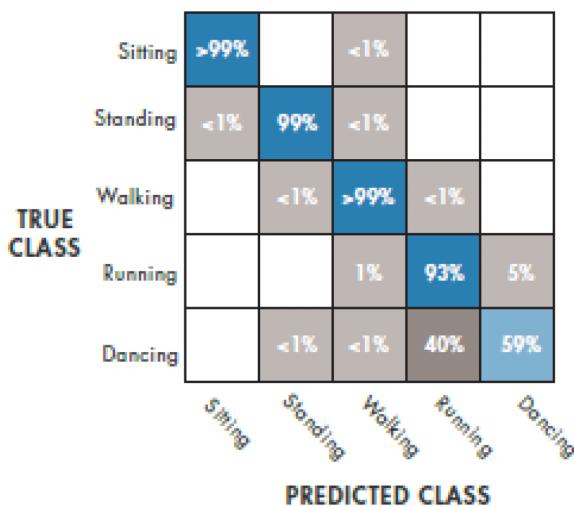
Abbildung 9: Common techniques used to derive features from data set

7.5.4 Step 4: Build and Train the Model

When building a model, it's a good idea to start with something simple; It will be faster to run and easier to interpret. We start with a basic decision tree.



To see how well it performs, we plot the confusion matrix:



It gives us a 94.1% correct classification rate on train. Anyway, it also shows that our model is clearly having trouble distinguishing between dancing and running. Maybe a decision tree doesn't work for this type of data. Then, we will try a few different algorithms.

K-nearest neighbors (KNN), a simple algorithm that compares new points to the training data and returns the most frequent class of the „K“ nearest points. That gives us 98% accuracy compared to 94.1% for the simple decision tree and a better confusion matrix.

	Sitting	<1%			
Sitting	>99%	<1%			
Standing	1%	99%	1%		
Walking		2%	98%		
Running		<1%	1%	97%	1%
Dancing		1%	1%	6%	92%
PREDICTED CLASS					

However, KNNs take a considerable amount of memory to run, since they require all training data to make a prediction. We try a linear discriminant model, but that doesn't improve the results. Finally, we try a multi-class support vector machine (SVM). The SVM does very well - we now get 99% accuracy.

	Sitting	<1%			
Sitting	>99%	<1%			
Standing	<1%	>99%	<1%		
Walking		<1%	>99%		
Running			<1%	98%	2%
Dancing		<1%	<1%	3%	96%
PREDICTED CLASS					

We achieved our goal by iterating on the model and trying different algorithms. If our classifier still couldn't reliably differentiate between dancing and running, we'd look into ways to improve the model.

7.5.5 Step 5: Improve the model

Improving a model can take two different directions: make the model simpler (simplify) or add complexity. First, we look for opportunities to reduce the number of features. Popular feature reduction techniques include:

- **Correlation matrix** - shows the relationship between variables, so that variables (or features), that are not highly correlated with the output, can be removed.
- **Principal component analysis (PCA)** - eliminates redundancy by finding a combination of features that captures key distinctions between the original features and brings out strong patterns in the dataset.
- **Sequential feature reduction** - reduces features iteratively on the model until there is no improvement in performance.

Next, we look at ways to reduce the model itself. We can do this by:

- Pruning branches from a decision tree
- Removing learners from an ensemble

The other approach is to add Complexity:

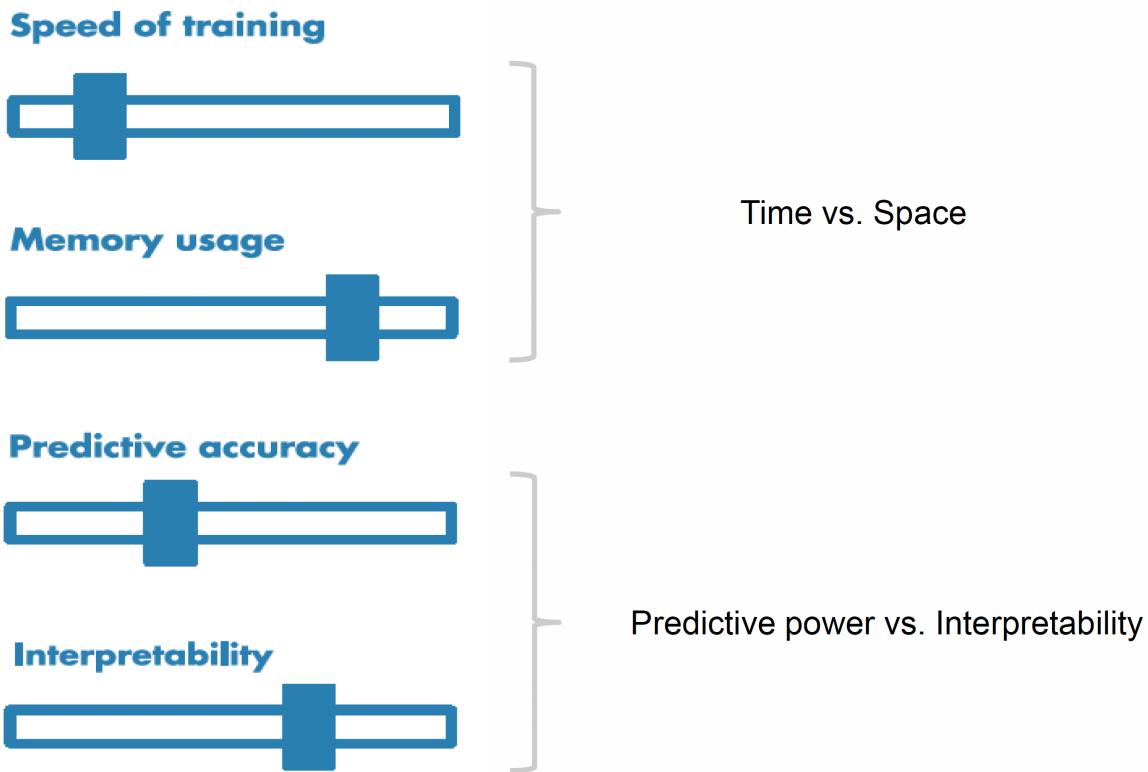
- If our model can't differentiate dancing from running because it is over-generalizing, then we need to find ways to make it more fine-tuned.

To do this, we can either:

- **Use model combination** — merge multiple simpler models into a larger model that is better able to represent the trends in the data than any of the simpler models could on their own.
- **Add more data sources** — look at the gyroscope data as well as the accelerometer data. The gyroscope records the orientation of the cell phone during activity. This data might provide unique signatures for the different activities; e.g. there might be a combination of acceleration and rotation that's unique for running.

Once we've adjusted the model, we validate its performance on the test data that we set aside during preprocessing. If the model can reliably classify activities on the test data, we're ready to move it to the phone and start tracking.

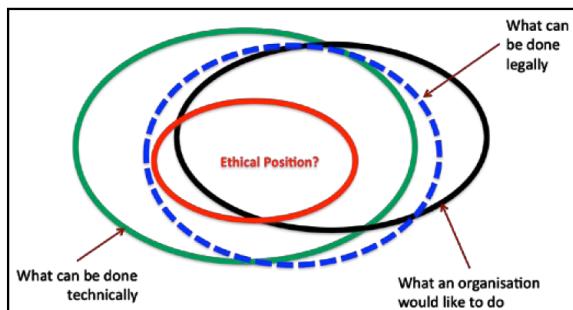
7.6 How to choose the right model (ML view)



8 Ethics in Data Science

8.1 Data Ethics

- Data ethics is an emerging branch of applied ethics.
- It describes the value judgements and approaches we make when generating, analysing and disseminating data.
- This includes:
 - a sound knowledge of data protection law and other relevant legislation
 - the appropriate use of new technologies
- UK Data Ethics Framework ([Link](#))



8.2 Big Data Analytics and Data Science

- Some of the distinctive aspects of big data analytics are:
 - the use of algorithms
 - the opacity of the processing
 - the tendency to collect „all the data“
 - the re-purposing of data
 - the use of new types of data
- Types of data:
 - **Provided data** is consciously given by individuals, e.g. when filling in an online form
 - **Observed data** is recorded automatically, e.g. by online cookies or sensors or CCTV linked to facial recognition
 - **Derived data** is produced from other data in a relatively simple and straightforward fashion, e.g. calculating customer profitability

8.3 Possible problems with data science

- Some types of big data analytics, such as profiling, can have intrusive effects on individuals
 - it may use personal data to make decisions affecting individuals
 - e.g. credit limits, but also promotions („people analytics“)
- Organisations need to consider whether the use of personal data in big data applications is within people's reasonable expectations
- The complexity of the methods of big data analysis, such as machine learning, can make it difficult for organizations to be transparent about the processing of personal data.

8.4 Ethical concerns about personal data

- **Surveillance**

A 2013 study by the Wellcome Trust, consisting of focus groups and telephone interviews, found there to be a „widespread wariness“ about being spied on by the government, corporations and criminals.

- **Discrimination**

The same study revealed concerns about possible discrimination against people based on medical data, for instance where such data is shared with employers who might make discriminatory decisions about people because of mental health issues.

- **Consent**

An online survey conducted by Demos in 2012 found that people's top concern for personal data use was about companies using it without their consent.

- **Data sharing**

The Institute for Insight in the Public Services conducted a telephone survey in 2008 which revealed that while people are generally happy for their personal data to be held by one organisation, they are concerned when it is shared with others. These concerns centered on loss of control over personal data and fears that errors in the data would be perpetuated through sharing.

8.5 Conditions for processing personal data

1. **Consent** must be a freely given, specific and informed indication that subjects agree to the data processing.
2. Data processing must be necessary for the **legitimate interests of the organization** collecting the data.
3. Data processing must be **necessary for the performance of a contract** to which the data subject is a party.

Big data analysis carried out in the public sector may be legitimized by other conditions, for instance where processing is necessary for the exercise of functions of a government department.

8.6 Purpose Limitation

- The purpose limitation principle does not necessarily create a barrier for big data analytics, but it means an **assessment of compatibility** of processing purposes must be done.
 - First, the **purpose** for which the data is collected must be specified and lawful.
 - Second, if the data is further processed for any other purpose, it **must not be incompatible with the original purpose**.
 - If an organization is buying personal data from elsewhere for big data analytics, it needs to **practice due diligence**. It will need to assess whether the new processing is incompatible with the original purpose for which the data was collected.
- **Fairness** is a key factor in determining whether big data analysis is incompatible with the original processing purpose.

8.7 Data Minimization: Collection & Retention

- Big data analytics can result in the collection of personal data that is excessive for the processing purpose.
- Organizations may be encouraged to retain personal data for longer than necessary because big data applications are capable of analyzing large volumes of data.

8.8 Accuracy

- There are implications regarding the **accuracy** of personal data at all stages of a big data project: collection analysis and application.
 - Personal data must be accurate and, where necessary, kept up to date
- Results of data analysis may not be **representative** of the population as a whole.
- **Hidden biases** in data sets can lead to inaccurate predictions about individuals.

8.9 Rights of Individuals

- The vast quantities of data used in big data analytics may make it more difficult for organisations to comply with the **right of access to personal data**.
- People have the right to be told:
 - What personal data about them is being processed.
 - The purposes for which it is being processed.
 - Who it may be disclosed to.
 - They have the right to receive a copy of the information that constitutes their personal data, as well as information about its sources.
- Organisations will need to have appropriate processes in place to deal with the GDPR's extension of **rights regarding decisions based on automated processing**.
- People have the **right not to be subject to a decision based solely on automated processing**, including profiling, if it „significantly affects“ them, such as automated decisions made on online credit applications, VISA emission or e-recruitemet.

8.10 Security

- There are several **information security risks** specific to big data analytics.
 - E.g. the high level of replication in big data storage and the frequency of outsourcing the analytics increase the risk of breaches, data leakages and degradation; also, the creation of links between the different datasets could increase the impact of breaches and leakages.
- Organisations need to recognize these new risks and put in place appropriate security measures.
(Link for Cloud Computing)

8.11 Accountability and Governance

- **Accountability** is increasingly important for big data analytics and will become an explicit requirement under the GDPR.
 - Records of processing activities must be maintained when processing personal data that could result in a risk to individuals' rights and freedoms.
- Big data organisations may need to **make changes to their reporting structures**, internal record keeping and resource allocation.
- Machine learning algorithms have the potential to make decisions that are **discriminatory, erroneous and unjustified**.
- **Data quality** is a key issue for those with information governance responsibilities in a big data context.

8.12 Data Controllers & Data Processors

- Big data analytics can make it difficult to distinguish between *data controllers* and *data processors*.
- A data controller has **overall control over the processing of personal data** - it determines the purposes and manner of the processing, either on its own, jointly or in common with another data controller.
- A data processor does not have such control - it merely **processes personal data on a data controller's behalf**.
- Organizations outsourcing analytics to companies specializing in AI and machine learning need to consider carefully who has control over the processing of any personal data.

8.13 Key Recommendations

8.13.1 Anonymization

- Carefully **consider** whether the big data analytics to be undertaken actually requires the processing of personal data.
- Often, this will not be the case; in such circumstances, organizations should use appropriate techniques to **anonymize** the personal data in their dataset(s) before analysis.

8.13.2 Privacy Notices

- Be **transparent** about their processing of personal data by using a combination of innovative approaches in order to provide meaningful privacy notices at appropriate stages throughout a big data project.
- This may include the use of icons, just-in-time notifications and layered privacy notices.
- The processing of personal data cannot be considered **fair** unless the data subject is given some basic information about it, including the data controller's identity, the purpose of the processing and any other information necessary.
- Ticking a box to say „I have read and agreed to the terms and conditions“ has been described as „**the biggest lie on the web**“.

8.13.3 Privacy Impact Assessment

- Embed a **privacy impact assessment framework** into the big data processing activities to help identify privacy risks and assess necessity and proportionality of a given project.
- The privacy impact assessment should involve **input from all relevant parties** including data analysts, compliance officers, board members and the public.

8.13.4 Privacy by Design

- Adopt a privacy by design approach in the development and application of their big data analytics.
- This should include implementing **technical and organizational measures** to address matters including *data security, data minimization and data segregation*.

8.13.5 Ethical Approaches

- Develop ethical principles to help reinforce key data protection principles.
- Employees in smaller organisations should use these **principles as a reference point** when working on big data projects. Conversely, larger organizations (e.g. IBM, Accenture are doing it) should create ethics boards to help scrutinise projects and assess complex issues arising from big data analytics.

8.13.6 Algorithmic Transparency

- Implement innovative techniques to develop **auditable** machine learning algorithms (→ explainable AI)
- Internal and external audits should be undertaken with a view to **explaining the rationale behind algorithmic decisions** and checking for bias, discrimination and errors.